wb_use

HELPFUL WAYBACK MACHINE INFO

Dave Sherfesee

01/26/2002

THE FULL NETWORK ARCHITECTURE OF THE WAYBACK MACHINE ASSIGNS SPECIFIC TASKS TO THE VARIOUS COMPUTERS WHICH SERVE THE DATA. IN FACT, EACH MACHINE WILL PERFORM ONE OF THREE DISTINCT TASKS. A COMPUTER WILL SERVE EITHER AS A CGI MACHINE, wb_server HOST, OR wb_tcp HOST. THESE THREE FUNCTIONS ARE ALL PERFORMED BY ONE MACHINE IF THE "simple" ARCHITECTURE IS USED (AS OPPOSED TO THE "network" ARCHITECTURE). HERE ARE A FEW POINTERS FOR CONFIGURING AND RUNNING THE WAYBACK MACHINE.

I. CGI MACHINE

1. LOCATION OF WAYBACK TREE: /alexa/apache/vhosts/archive

2. LOCATION OF APACHE CONFIG FILE: /alexa/apache/conf/httpd.conf

   THIS FILE NEEDS TO BE EDITTED IF THE NAME OF THE CGI MACHINE CHANGES. THE FIELDS THAT MUST BE EDITTED ARE THE "ServerName" APACHE FIELD AND THE "RedirectMatch" FIELDS IN THE VIRTUAL HOST DIRECTIVE.

3. STARTING APACHE: /alexa/apache/bin/apachectl start|stop|graceful|restart

   RUN AS ROOT. BECAUSE THE WAYBACK MACHINE RUNS UNDER MOD_PERL, APACHE SHOULD BE RESTARTED AFTER ANY CHANGES ARE MADE TO THE CODE. THIS ISN'T NECESSARY, BUT SINCE APACHE CACHES THE CGI SCRIPTS, CODE CHANGES WON'T BE SEEN UNTIL EACH APACHE PROCESS HAS DIED AND BEEN RESTARTED.

4. APACHE LOGS: /alexa/apache/logs -> /export/logs

   BOTH ERROR AND ACCESS LOGS WILL BE LOCATED HERE.

5. WAYBACK CACHE LOCATION: /alexa/apache/vhosts/archive/live_dir
   /alexa/apache/vhosts/archive/db_dir

   THE CACHE OF DOCUMENTS RETRIEVED FROM THE LIVE WEB AND ARCHIVE WILL BE STORED IN THESE TWO DIRECTORIES. THE IN-PROGRESS ARC FILES WILL RESIDE IN live_dir, AND THE LOOK-UP TABLES (DBM FILES) IN db_dir. THESE DIRECTORIES MUST BE OWNED AND WRITEABLE BY THE APACHE USER.

6. CONFIGURATION FILE: /alexa/apache/vhosts/archive/cgi-bin/wayback.cgi

   THIS IS THE FILE WHICH CONFIGURES THE WAYBACK MACHINE AND EXECUTES EACH INCOMING REQUEST. TURN THE FOLLOWING OPTIONS ON BY ASSIGNING '1' AND OFF BY ASSIGNING 'undef'. WHERE THESE OPTIONS DON'T MAKE SENSE, USE 'undef' FOR THE DEFAULT BEHAVIOR (EXAMPLE: $MAX_N_RECORDS). A DESCRIPTIOIN SHOULD ACCOMPANY EACH OF THESE VARIABLES EARLIER IN THE FILE.

```
my $DEBUG                   = 0;       # '1' FOR DEBUG MESSAGES, '0' ELSE
my $TIME                    = 0;       # '1' FOR TIMING MESSAGES, '0' ELSE
my $DEFAULT_COLLECTION      = undef;   # COLLECTION IF NONE PROVIDED IN URL
my $DISABLE_EXCLUDE_FILE    = 1;       # '1' TO DISABLE EXCLUDE FILES
my $DISABLE_LIVE_WEB        = 1;       # '1' TO DISABLE LIVE WEB RETRIEVALS
my $DISABLE_REDIRECTS       = undef;   # '1' TO DISABLE REDIRECTS
my $DISABLE_ROBOTS          = undef;   # '1' TO DISABLE ROBOTS
```

wb_use

```
my $DO_ROBOTS_IP_LOOKUPS    = undef;    # '1' TO FIND ROBOTS IP ADDRESSES
my $ENFORCE_USER_AGREEMENT  = undef;    # '1' TO ENFORCE A USER AGREEMENT
my $MAX_N_REDIRECTS         = undef;    # NUMBER OF REDIRECTS TO FOLLOW
my $MAX_N_RECORDS           = undef;    # NUMBER OF QUERY RECORDS TO RETURN
my $MAX_ROBOTS_DURATION     = 84600;    # NUMBER OF SECONDS TO USE ROBOTS
RULES

my $MODE                    = "network"; # "network" FOR FULL ARCHITECTURE
my $NETWORK_REQUEST_MODE    = undef;    # "pops" FOR pops "network"
my $USE_WEB_FOR_ROBOTS      = undef;    # '1' TO GETS ROBOTS FROM WEB
my $USE_BANNER              = undef;    # '1' TO USE A BANNER
my $URL_PURIFY_DIRECTORY    = undef;    # DIRECTORY WITH URL PURIFIY FILES
```

BECAUSE THE WAYBACK MACHINE RUNS UNDER MOD_PERL, APACHE SHOULD BE RESTARTED AFTER ANY CHANGES TO THE CODE.

7. NETWORK wb_server CONFIG FILE:

   /alexa/apache/vhosts/archive/cgi-bin/wb_network_alpha.conf

   THE CGI MACHINE READS THIS FILE TO DETERMING WHICH wb_server TO QUERY FOR THE REQUESTED DATA. IF THE NAME OF THE wb_server HOST CHANGES, OR IF NEW wb_server HOSTS ARE ADDED, THIS FILE MUST BE EDITTED TO REFLECT THESE CHANGES.

8. PERL LIBRARIES:

   a. BUILD PERL WITH LARGE-FILE SUPPORT
   b. INSTALL THE FOLLOWING MODULES (OR MORE RECENT VERSIONS OF THEM):

      1. Time-HiRes-01.20
      2. MIME_Base64-2.12
      3. URI-1.18
      4. HTML-Tagset-3.03
      5. HTML-Parser-3.25
      6. Compress-Zlib-1.16
      7. libnet-1.10
      8. Digest-MD5-2.16
      9. libwww-perl-5.64
      10. Net-DNS-0.14
      11. DB_File-1.803
      12. mod_perl-1.26

II. wb_server HOST

1. wb_server TREE: /alexa/wb_server

2. CONFIG FILE: /alexa/wb_server/wb_server.conf

3. STARTING THE SERVER:

   /alexa/wb_server/wb_server_daemon (start|stop|restart)
   RUN AS ROOT.

4. PATH FILE NOTE:

   IF THE NAME OF THE wb_tcp HOST CHANGES, OR IF ANY ARC FILES ON wb_tcp HOSTS ARE MOVED, THE PATH FILE MUST BE MODIFIED/REGENERATED TO REFLECT THE NEW NETWORK PATH TO THE ARC FILES.

III. wb_tcp HOST

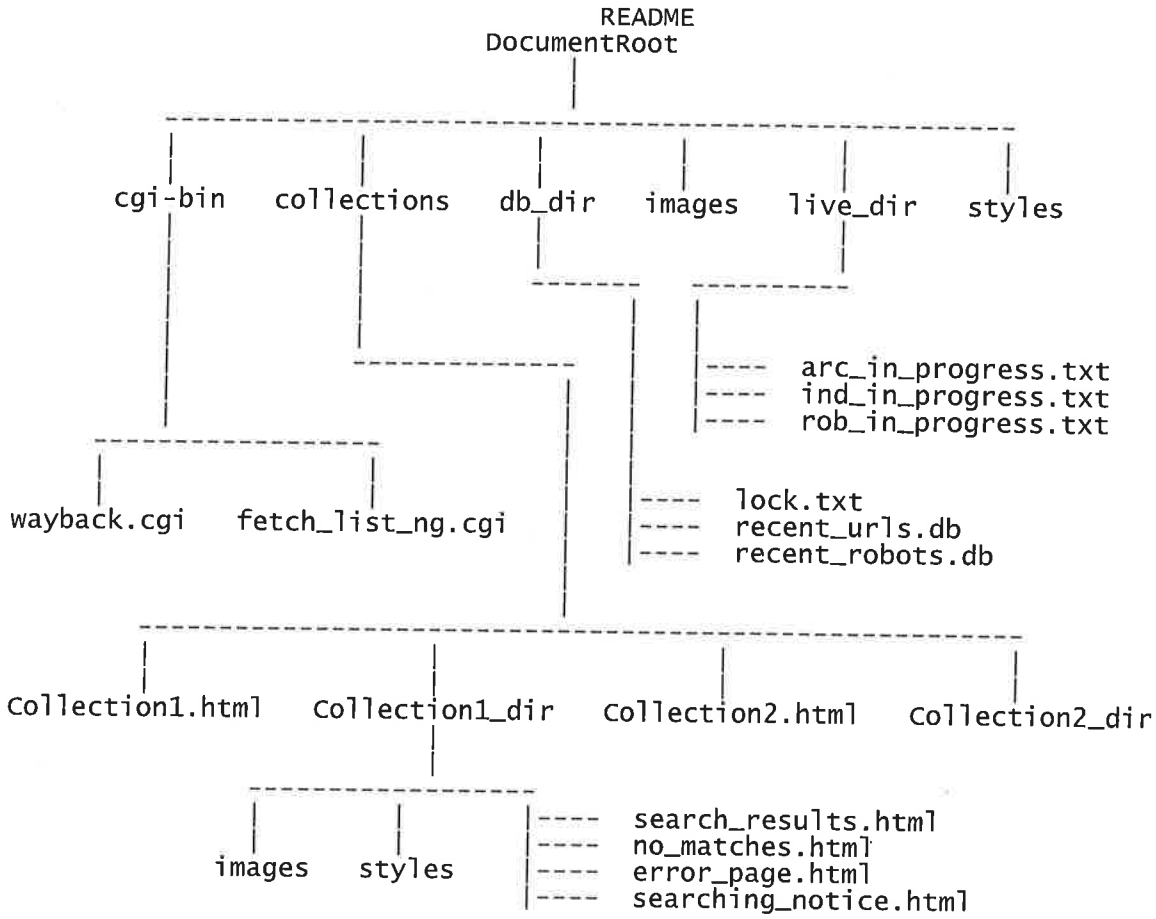
1. wb_tcp TREE: /alexa/wb_tcp

2. STARTING THE SERVER:

/alexa/wb_tcp/wb_tcp_daemon (start|stop|restart)
RUN AS ROOT. THIS SERVER SHOULD START AT BOOT TIME.

3. wb_tcp NOTE:

THE wb_tcp SERVER WILL START AT BOOT-TIME. IT ANSWERS REQUESTS FROM THE wb_server FOR LOCAL DOCUMENTS, RETRIEVES THEM FROM ARC FILES, AND RETURNS THEM SO THAT THE wb_server CAN RETURN ARCHIVED DOCUMENTS TO THE CGI MACHINE.



The simplest way to configure the web server (apache) is to use a virtualhost directive.

```
<VirtualHost *>
    ServerName archive.alexa.com
    ServerAdmin servers@alexa.com
    DocumentRoot /alexa/apache/vhosts/archive
    CustomLog /export/logs/archive-access_log alexa

# WAYBACK MACHINE
# THE ORDER OF THE FOLLOWING ALIASES IS IMPORTANT.

    AliasMatch ^$  /alexa/apache/vhosts/archive/index.html
    AliasMatch ^/$ /alexa/apache/vhosts/archive/index.html

    RedirectMatch ^/e2k/*$ http://archive.alexa.com/collections/e2k.html
    RedirectMatch ^/web/*$ http://archive.alexa.com/collections/web.html

    Alias /wayback /alexa/apache/vhosts/archive
    Alias /collections /alexa/apache/vhosts/archive/collections
    Alias /images /alexa/apache/vhosts/archive/images
    Alias /robots.txt /alexa/apache/vhosts/archive/archive_notices/robots.txt

    ScriptAlias /archive_request_ng
/alexa/apache/vhosts/archive/cgi-bin/fetch_list_ng.cgi

# SEND URLS TO THE WAYBACK MACHINE SCRIPT.

    ScriptAliasMatch /(web\/.+) /alexa/apache/vhosts/archive/cgi-bin/wayback.cgi/$1
```

currently set to machine IP

```
    ScriptAliasMatch /(e2k\/.+) /alexa/apache/vhosts/archive/cgi-bin/wayback.cgi/$1
    ScriptAliasMatch /(.+) /alexa/apache/vhosts/archive/cgi-bin/wayback.cgi/$1

</VirtualHost>
```

We strongly encourage the use of mod_perl when running the Wayback Machine, as WayBack.pm is a large and complex perl module, and the overhead of the perl interpreter is significant when launched on each request. If you have mod_perl installed, include the following directive in your virtualhost block:

```
   <IfModule mod_perl.c>

      <Location "/archive_request_ng">
         SetHandler perl-script
         PerlHandler Apache::Registry
         Options ExecCGI
         allow from all
         PerlSendHeader On
         PerlSetEnv CGI_NAME "fetch_list_ng"
      </Location>

      <Location "/$ARC_URL_PREFIX">
         SetHandler perl-script
         PerlHandler Apache::Registry
         Options ExecCGI
         allow from all
         PerlSendHeader On
         PerlSetEnv CGI_NAME "WAYBACK MACHINE"
      </Location>
```

and/or

```
      <Location "/collection1">
         SetHandler perl-script
         PerlHandler Apache::Registry
         Options ExecCGI
         allow from all
         PerlSendHeader On
         PerlSetEnv CGI_NAME "WAYBACK MACHINE"
      </Location>

            .
            .
            .

      <Location "/collectionN">
         SetHandler perl-script
         PerlHandler Apache::Registry
         Options ExecCGI
         allow from all
         PerlSendHeader On
         PerlSetEnv CGI_NAME "WAYBACK MACHINE"
      </Location>

      PerlWarn On
      PerlTaintCheck On

   </IfModule>

##########################################################################
#                      WAYBACK MACHINE CONFIGURATION                     #
##########################################################################
```

The Wayback Machine perl module, WayBack.pm exports a number of configuration methods. If used, they must be employed before handling any requests. Only the configure and build_cdx_hash methods **MUST** be called, and the build_cdx_hash need not be called if using the "network" architecture. Here is a list of the configuration methods exported by WayBack.pm:

```
disable_live_web ();        # Do not retrieve documents from the live web.
                            # The default behavior is to retrieve requested
                            # documents from the live web if they do not
                            # exist in the collection.

disable_redirects ();       # Do not return redirects with alternate dates.
                            # The default behavior is to return a redirect if
                            # the date of the document to be returned does not
                            # exactly match the request date.  This way, the
                            # user knows the date of the document returned.

disable_exclude_file ();    # Do not filter domains with an exclude file.
                            # The exclude file is used to filter out URL's
                            # for domains which have requested not to be
                            # archived.  The default behavior is to use this
                            # file to exclude URL's.

disable_robots ();          # Do not filter documents with robots.txt rules.
                            # The default behavior is to filter URL's based
                            # on robots.txt rules.

enable_debug_msgs ();       # Turn on debugging.  This will result in some
                            # messages being written to stderr.

enable_timing_msgs ();      # Turn on profiling.  This aids in determining
                            # where most of the time is spent during a
                            # request.

use_web_for_robots ();      # Retrieve robots.txt documents from the web.
                            # The default behavior is to retrieve robots.txt
                            # documents from the archive.

do_robots_ip_lookups ();    # Look up ip addresses for robots.txt documents.
                            # If robots.txt documents are retrieved from the
                            # live web, they are archived for addition to the
                            # collection.  IP-addresses are part of the
                            # arc-file meta-data.  IP lookups effect
                            # performance, so they are not done by default.
```

set_default_collection:

The name of the default data collection is "web". To change this name, pass the new default collection to this method.

```
   set_default_collection ($DEFAULT_COLLECTION);
```

set_max_robots_duration:

The robots.txt documents are cached in a dbm file when they are used. This allows for speedier retrieval on the next request for this document. The default time period to keep the robots.txt document cached is one week. After this time period, a request for this document will require a new retrieval from either the archive or the live web, depending on the robots configuration. To change the default robots.txt caching duration, pass a value to this method, the time period specified in units of seconds.

```
    set_max_robots_duration ($MAX_ROBOTS_DURATION);
```

set_max_n_redirects:

Many archived documents contain redirects to other urls, either because of "Location: url" HTTP headers, content refreshed, or some other mechanism. The default behavior is to follow 3 redirects and then fail. To change the number of redirects to follow, pass the value to this method.

```
   set_max_n_redirects ($MAX_N_REDIRECTS);
```

set_max_n_records:

There are many copies of some documents. We don't want to return 10,000 records to the user because it would burden their browser to have to load and render such a page. The default number of records to return is 1000. To change the maximum number of records to return, pass the value to this method.

```
   set_max_n_records ($MAX_N_RECORDS);
```

set_mode:

The Wayback Machine supports two architectures: "simple" and "network". In the "simple" architecture the WayBack.pm module (along with supporting perl modules) constitutes the whole of the Wayback Machine. The "network" architecture is employed by Alexa and should only be utilized if:

1. The archive is very large, utilizing a distributed cdx file system
2. Many different collections are supported
3. One has significant systems administration/operations resources

The "network" architecture is detailed in a technical specification on the Alexa site and is employed as a 4-tier server, utilizing many machines. The default architecture, or mode, is "simple". To enable the "network" architecture, pass that string to this method.

```
   set_mode ($MODE);
```

configure:

This is the only configuration method which **MUST** be called in the perl cgi which imports WayBack.pm. This method must be passed all of the files which will be used by the Wayback Machine as well as some information about the archival URL it should expect to parse.

%COLLECTION_HASH is a hash of hashes. The hash keys are the possible collections which can be specified in the archival URL, and the value of each possible collection must be an array which specifies the files to use:

```
   $NO_RESULT_FILE      # html template for unsuccessful query results
   $RESULT_FILE         # html template for successful query results
   $ERROR_FILE          # html template for error message
   $SEARCHING_FILE      # html for "Searching the Archive" notice
```

Example: If we support a "web" collection and a "e2k" collection, we might declare:

```
my %COLLECTION_HASH = (
   "web" => [
             "../collections/web/no_matches.html",
             "../collections/web/search_results.html",
             "../collections/web/error_page.html",
```

```
                "../collections/web/searching_notice.html",
              ],
    "e2k" => [
                "../collections/e2k/no_matches.html",
                "../collections/e2k/search_results.html",
                "../collections/e2k/error_page.html",
                "../collections/web/searching_notice.html",
              ],
);
```

@CACHE_FILES is a list of files used in the caching/archiving of data. This list must include (in order) the following files:

```
    $ARC_IN_PROGRESS_FILE     # arc-file to build from
                              # live-web-retrieved documents

    $ROB_IN_PROGRESS_FILE     # arc-file to build from live-web-retrieved
                              # robots.txt documents

    $DB_LOCK_FILE             # dummy file for flock file locking

    $DBM_RECENT_URLS_FILE     # berkeley db in which are stored
                              # live-web-retrieved documents

    $DBM_RECENT_ROBOTS_FILE   # berkeley db in which are stored recently
                              # retrieved robots.txt documents
```

An "undef" file entry disables this feature. An example decalaration might be:

```
my @CACHE_FILES = (
                    "../live_dir/arc_in_progress.txt",
                    "../live_dir/rob_in_progress.txt",
                    "../db_dir/lock.txt",
                    "../db_dir/recent_robots.db",
                    "../db_dir/recent_urls.db",
                  );
```

@EXCLUDE_FILES is a list of files used for doing additional data filtering. The necessary elements are:

```
   $EXCLUDE_FILE      # sorted list of domains to exclude
   $C_EXCLUDE_FILE    # sorted list of canonized domains to exclude
```

AN "undef" FILE ENTRY DISABLES THIS FEATURE. An example declaration might be:

```
my @EXCLUDE_FILES = (
                     "/net/arc42/0/CRAWL/name_excludes",
                     "/net/arc42/0/CRAWL/c_name_excludes",
                    );
```

$ARC_URL_PREFIX is the URL path which signifies that the url is an archival URL. The archival URL is detailed in a technical specification on the Alexa site. Basically, if you configure your Apache web server to recognize that urls such as http://yourdomain.com/wayback/... signifies an archival url, then you would want to set:

```
my $ARC_URL_PREFIX = "wayback/";
```

If all urls on that domain (maybe you're using virtualhosts) are to be archival urls, set $ARC_URL_PREFIX = "".

%IMAGE_HASH is a hash which specifies the images to use to denote page retrieval speeds. The hash values should be static URL paths.

```
my %IMAGE_HASH = (
                  "fast" => "/wayback/images/fast.gif",
                  "okay" => "/wayback/images/okay.gif",
                  "slow" => "/wayback/images/slow.gif",
);
```

%PATH_HASH is a hash of arrays. It lists all of the collections served and the path files which belong to each collection. This is used only with the "simple" architecture. Example:

```
my %PATH_HASH = (
   "web" => [
             "/alexa/wayback/path/path.txt",
             "/alexa/wayback/path/path.txt",
            ],
   "e2k" => [
             "/alexa/wayback/path/e2k_path1.txt",
             "/alexa/wayback/path/e2k_path2.txt",
            ],
);

   configure (\%COLLECTION_HASH,
              \@CACHE_FILES,
              \@EXCLUDE_FILES,
              $ARC_URL_PREFIX,
              \%IMAGE_HASH,
              \%PATH_HASH);
```

build_cdx_hash:

This method constructs the hash of cdx header information for all of the cdx files used. It only applies to the "simple" architecture.

%CDX_HASH is a hash of arrays. It lists all of the collections served and the cdx files which belong to each collection. This is used only with the "simple" architecture.

```
my %CDX_HASH = (
   "web" => [
             "/alexa/wayback/cdx/web_cdx1.cdx",
             "/alexa/wayback/cdx/web_cdx2.cdx",
            ],
   "e2k" => [
             "/alexa/wayback/cdx/e2k_cdx1.cdx",
             "/alexa/wayback/cdx/e2k_cdx2.cdx",
            ],
);
```

Pass this hash to the build_cdx_hash method.

```
   build_cdx_hash (\%CDX_HASH);
```

```
##############################################################################
#                                INSTALLATION                                #
##############################################################################
```

To install WayBack.pm, you should be able to issue the following commands:

```
   perl Makefile.pl
   make
```

```
   make test
   make install

The Wayback Machine requires the following packages, available on CPAN:

DB_File;                    # database management module
HTTP::Request;              # HTTP request utilities
LWP::UserAgent;             # LWP wrapper for HTTP request
Net::DNS;                   # DNS utilities
Time::Local;                # date manipulation routines
Time::HiRes;                # benchmarking module
Fcntl ":flock";             # allow for file locking
Compress::Zlib;             # compression utilities

The Wayback Machine also requires the following Alexa Internet modules:

BinSearch;                  # binary search utilities
UrlPurify;                  # URL purification utilities

Note that UrlPurify requires three text files, name_canon.txt, url_clean.txt,
and commerse_servers.txt to be located in the directory /alexa/url.  This
can be reconfigured in the UrlPurify module with the set_canon_directory ()
method, but this has not been built into WayBack.pm.

##############################################################################
#                                    END                                     #
##############################################################################
```

README FOR ALEXA INTERNET WAYBACK MACHINE

This document describes the functionality, configuration, and installation of the Wayback Machine, built by Alexa Internet. The Wayback Machine is an interface to an archive of internet documents. It allows users to query the archive for documents matching a URL and other optional criteria, retrieves documents from the archive, and attempts to reconstruct and render the document as it would have appeared when it was originally archived. Documents can be filtered in several ways, including robots.txt exclusions, and most of the behavior can be configured for various purposes.

Engineer and Author: Dave Sherfesee
Alexa Internet

ARCHIVE

The Wayback Machine is an interface to a collection of internet documents (web pages, images, javascript files, etc.). These documents must be stored in arc-file format, indexed in cdx-file format, and referenced in a path file. Technical specfications for arc files and cdx files reside on the Alexa Internet web site (http://www.alexa.com), but here is a brief description.

Arc File:

An arc file (archive file) is an ascii text file -- typically 100 Mbytes in size -- which contains the archived documents. An arc file is formatted as follows:

meta-data line
archived document

meta-data line
archived document

meta-data line
archived document

where the "meta-data line" is a " " delimited line containing five strings:

URL IP-address Archive-date Content-type Archive-length

URL is the URL of the archived document; IP-address is the IP address of the archived document; Archive-date is the 14-digit timestamp describing the time at which the document was archived (YYYYMMDDHHMMSS); Content-type is the MIME type of the document; and Archive-length is the length (in bytes) of the archived document. All documents are separated from the next meta-data line by a newline '\n' character. Each arc file begins with a header describing the arc-file version and format, and I encourage you to read the Arc File Technical Specification on the Alexa Internet site.

Alexa employs a proprietary compression scheme on the arc files, allowing for random access into the compressed file. This compression scheme must be used on the arc files for the "simple" Wayback architecture. The "network" architecture also allows for uncompressed documents. Straight gzip is no longer supported due to performance costs.

CDX File:

A cdx file is an index file describing the content of the archive. The Wayback Machine uses the cdx files to determine what documents exist in the archive as well as their location in the archive. A cdx file is a sorted file, each line containing a " " delimited record of an archived document. The first line of a cdx file must be a cdx header, describing the format of the cdx file. There are many required fields in the cdx file, but they can be in any order, with the exception of the first field, which must be the canonized URL. Here is an example cdx header and record:

```
 CDX A b e a m s c k r V v D d g M n
somedomain.com/images/image1.jpg 20000305225258 000.000.000.000
www.somedomain.com:80/images/image.jpg image/jpeg 200
16d76ab0e3e2d38e1c4b8a5504339fe7 16d76ab0e3e2d38e1c4b8a5504339fe7 - 23163754
62630679 2687031 9530122 arc_file_path - 33895
```

The cdx file header has the format " CDX ? ? ? ? ..." where each "?" represents a character. The character describes the data in that corresponding column. For instance, the cdx header above begins with the fields "A b e a m" which represent:

A: canonized URL
b: Archive-date
e: IP-address
a: original URL
m: Content-type

These fields, along with:

s: HTTP Response Code
c or k: checksum
r: Redirect URL
V or v: Compressed or uncompressed offset of document in arc file
g: path to arc file
n: Content-length

are required by the Wayback Machine. Once the applicable records for a URL are located in a cdx file, documents may be retrieved from the arc files by searching for the "path to arc file (g)" in a path file.

Cdx files must be uncompressed in the "simple" Wayback architecture. The "network" architecture supports Alexa compression, but this is not recommended if performance is an issue. Straight gzip is not supported.

Path File:

A path file is simply a sorted ascii text file, each line being a " " delimited record of the arc files in the collection. Each line has the format:

path-to-arc-file network-path /dev/null

path-to-arc-file is the string found in the cdx file, network-path is the full path to the arc file on the network, and /dev/null is the path to another file (used internally by Alexa, but not necessary here). Path files are used so that arc files can be relocated within the network without having to recreate the cdx file. Path file recreation is simple and fast.

Path files must be uncompressed.

In general then, a typical document retrieval goes as follows:

1. Search the cdx file for the record that best matches the search criteria.
2. Use the arc file path in the cdx record to search the path file.
3. Locate the arc file with the full network path found in the path file.

4. Retrieve the archived document using the offset and content-length data from the cdx record.

```
##############################################################################
#                                 FUNCTIONALITY                              #
##############################################################################
```

Here are the methods WayBack.pm exports for request handling and profiling.

Request Handling:

```
parse_archival_url ();              # Parse the archival URL.  This method
                                    # retrieves data from the archival URL
                                    # and assigns it to the wayback object.

purify_url ();                      # This routine purifies and canonizes
                                    # the url contained in $self->{URL} and
                                    # assigns the results to
                                    # $self->{PURE_URL} and
                                    # $self->{CANON_URL}.

handle_request (\$output);          # This routine handles the request,
                                    # filling $output with the response
                                    # which is to be returned to the client.

print_wayback_values ();            # This routine writes the values stored
                                    # in the wayback object.

handle (method_call);               # This routine is used to wrap method
                                    # calls, trap errors, and detect output.
```

Profiling:

```
note_method_start ("method_name"); # Note start of method "method_name".
note_method_end ("method_name");   # Note end of method "method_name".
write_timing_msgs ();              # Print the profiling data for the
                                   # request.
```

```
##############################################################################
#                                   INTERFACE                                #
##############################################################################
```

Archival URL:

The Wayback Machine is accessed via the archival URL. The archival URL is described in detail on the Alexa web site, but here is a brief description:

http://yourdomain.com/${ARC_URL_PREFIX}datespec(request_flags)/url

As described in the WAYBACK MACHINE CONFIGURATION section, $ARC_URL_PREFIX is a string which can be used in your Apache (or other) web server configuration file in order to recognize archival URLs. For instance, you might designate all urls whose paths begin with wayback/ to be archival URLs. In this case, your archival URL would look like:

http://yourdomain.com/wayback/datespec(request_flags)/url

If all yourdomain.com URLs are to be archival URLs, set $ARC_URL_PREFIX = "":

http://yourdomain.com/datespec(request_flags)/url

The datespec helps to describe the type of request being made. Briefly:

2000	Most recent document from the year 2000
20000101125555	Document closest to Jan 01, 2000 at 12:55:55
1999*	All documents from 1999
1999-2001	Document between 1999 and 2001

The request flags supported are built into the UI. Some of them are:

h_	Filter results based on exact hostname.
ta_	Place all results in a table, not just exact matches.
sa_	Show all results. Don't filter out duplicates.

Others will be implemented as the UI evolves.

The url at the end of the archival URL is taken to be the request URL. If there was no '*' in the datespec, this URL is retrieved from the archive. If there was a '*' in the datespec, the URL can end in "*", in which case, the result will be a list of all URLs matching the datespec and the request URL up to the '*'. For instance, an archival URL of

http://yourdomain.com/1999*/somedomain.com/*

might return the following URLs:

somedomain.com/
somedomain.com/images/
somedomain.com/images/image1.jpg

since all of these URLs begin with "somedomain.com/".

Files:

There are 4 files required by the Wayback Machine for the UI, and they are described in the WAYBACK MACHINE CONFIGURATION section. They are:

```
$NO_RESULT_FILE     # html template for unsuccessful query results
$RESULT_FILE        # html template for successful query results
$ERROR_FILE         # html template for error message
$SEARCHING_FILE     # html for "Searching the Archive" notice
```

The Wayback Machine inserts the request output into these templates and returns them to the client. These templates must include the appropriate strings for data substitution. You should investigate the example UI files provided and look for strings like URL_TO_SUBSTITUTE, NUMBER_TO_SUBSTITUTE, and LIST_TO_SUBSTITUTE.

```
##############################################################################
#                              APACHE CONFIGURATION                          #
##############################################################################
```

Some web server configuration is required for the Wayback Machine. The default UI is set up as follows:

Collection Front Pages:

http://yourdomain.com/collections/somecollection.html

Example:

http://yourdomain.com/collections/web.html
http://yourdomain.com/collections/e2k.html

Directory Structure: